

Available online at www.sciencedirect.com**ScienceDirect**

Energy Procedia 49 (2014) 2454 – 2461

Energy
Procedia

SolarPACES 2013

A novel approach to reduce ray tracing simulation times by predicting number of rays

S.J. Bode^{a*}, P. Gauché^a, D. Griffith^b^a*Department of Mechanical and Mechatronic Engineering, University of Stellenbosch, Matieland 7602, South Africa,*^b*Council for Industrial and Scientific Research, Pretoria, 0184, South Africa*

Abstract

A ray tracer, specifically designed for concentrating solar power systems, has been developed at the Solar Thermal Energy Research Group (STERG). In most ray tracers a simulation runs until a number of rays, predefined by the user, have been traced. In a novel approach the raytracer uses a user defined tolerable error and confidence interval to predict and automatically trace the required number of rays for a simulation. This reduces computational time as only the minimum required number of rays are traced. This method has not been incorporated into any of the openly available ray tracers. The method and algorithm incorporated into the STERG raytracer are described here. As a validation test case the Eurodish and optimised field layout of Planta Solar 10 plant have been simulated.

© 2013 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license

(<http://creativecommons.org/licenses/by-nc-nd/3.0/>).

Selection and peer review by the scientific conference committee of SolarPACES 2013 under responsibility of PSE AG.

Final manuscript published as received without editorial corrections.

Keywords: Monte Carlo; ray tracing

1. Introduction

Numerical codes have played a critical role in the research and design of CSP systems. Several codes have been developed which assist for both design and optimization of a CSP systems [1,2]. These codes can be grouped into two categories: process codes, which are used model the power generation process and optical codes, which are codes used for predicting radiative flux on a receiver. Restricted by the computational resource at the time, many of

* Corresponding author. Tel.: +27-74-130-9442

E-mail address: bodesj@gmail.com

the first generation optical codes used simplified techniques to calculate the total flux on a target. However, with the advancement in computational power, Monte Carlo ray tracing methods are becoming more prominent in CSP simulations.

Monte Carlo ray tracing codes are the preferred numerical simulation techniques due to their flexibility and their ability to deal with complex geometries. In Monte Carlo ray tracing solar radiation is modelled as a bundle of rays and the stochastic paths of the rays are traced as they interact with objects within the scene. Rays can either be traced from source to receiver [3,4] or rays can be generated on surface of the concentrator [5]. Traditionally ray tracing has been slow, with computational times proportional to the number of rays used in a simulation. However, as each ray is strictly independent, Monte Carlo ray tracing techniques adapted very well to parallelization [6], another reason why they have become increasingly popular for CSP research.

A collision-based Monte Carlo ray tracer (rays are traced from the source) has been developed at the Solar Thermal Energy Research Group (STERG) at Stellenbosch University. The ray tracer has been specifically designed for use within a solar thermal research environment to aid and further research in CSP simulations and design. Written in C++ the ray tracer has been developed with a robust foundation to ensure ease of extendibility and to allow for continuous development. With the multi-disciplinary work conducted at STERG it is likely that the end user may not be an optics specialist and so the code has been developed to be as intuitive as possible in order to be made available to any CSP researcher. This includes a method for estimating the number of rays required in a simulation.

Most ray tracing simulations are run until a predefined number of rays (samples) have been traced. For a non-optics specialist the number of rays is an abstract concept and a common assurance is *more is better*. This can lead to a long computational times as a needlessly large number of rays being traced. On the contrary, too few rays can produce erroneous results. The accuracy of a ray tracing simulation is dependent on the number of rays traced and differs for every scene. As mentioned, the computational time is also dependent on the number of rays being traced, which leads to the question “How many rays is enough?”

The statistical theory on which the method is based is provided in Section 2. The algorithm which has been incorporated into the STERG raytracer is described in Section 3. The results of some validating test cases are provided in Section 4 followed by a discussion in Section 5.

Nomenclature

E	error
N	number of rays
n	number of repeated simulations
s	standard deviation
t	value for the t-distribution
\bar{y}	mean estimate of the solution
y	solution to a simulation
z	normal distribution
α	confidence coefficient
σ	variance

2. Theory

To reduce ray tracing computational times a number of acceleration techniques have been developed [8]. Included in these techniques are variance reduction methods which improve Monte Carlo efficiency [9]. The method discussed in this paper is not a new method for reducing number of rays, but rather a method which, for given a ray tracing algorithm, estimates the minimum required number of rays needed to resolve the solution to a simulation.

This section provides some statistical theory used to predict the number of rays required to obtain an accurate

solution to a simulation. Here, the term *solution* refers to both the total power and the spatial resolution of radiant flux on a target if an infinite number of rays were to be traced. It should be noted though, that even if an infinite number of rays could be traced, a ray tracer can never fully simulate a real CSP system as every input into the ray tracer is a potential source of error, for example an imprecise description of the reflecting surface. In the STERG raytracer most of these errors are modelled using empirical or theoretical models and distributions. This paper deals with only one type of error known as shot noise.

First discovered by Walter Schottky [10], shot noise arises in the ray tracing due to the arrival of a discrete number of independent and randomly generated rays. In ray tracing the signal to noise ratio (SNR) of a result is proportional to the square route of the number of rays which have struck the target. Therefore a large SNR requires a large number of rays. Shot noise is observed as fluctuations in the results of a repeated simulation. This is illustrated in Fig. 1 which is the results of repeating the same simulation on a random field of 200 flat heliostats. Holding all parameters constant, Fig. 1(a) is the results of the simulation repeated 30 times using 1000 rays and Fig. 1(b) is the simulation repeated 30 times using 10 000 rays.

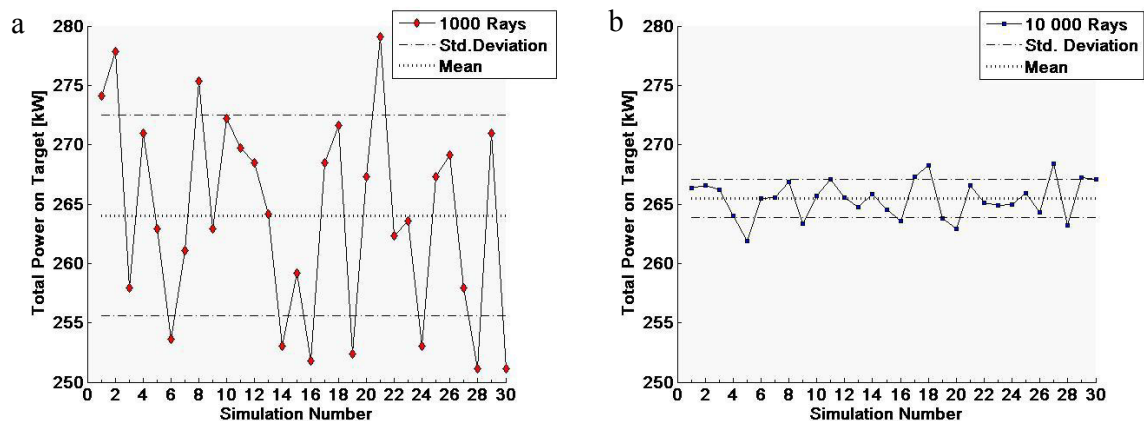


Fig 1. Results of a ray tracing simulation (a) using 1000 rays; (b) 10 000 rays

The development of the method is based on the concept that because each ray can be traced independently from the next, running a simulations with infinite number of rays ($N = \infty$) is equivalent to repeating a simulation with a finite number of rays ($N \geq 1$) infinite times ($n = \infty$). Stating the approach differently makes it possible to use statistical tools to make inferences about the minimum number of rays required to reduce the error in a solution to an acceptable level.

From Fig. 1 if a simulation with a finite number of rays is repeated n times, where $n > 0$ the results of each simulation will fluctuate about a certain mean value μ_N . As n is increased ($n \rightarrow \infty$) the mean, will tend toward the correct solution ($\mu_N \rightarrow \mu_\infty$) with a variance σ_N . Thus the solution to a simulation can be found if an adequate estimate of μ_∞ can be made.

This concept is illustrated in Fig. 2 where a simple flat 1m^2 heliostat with a reflectance of 0.97, rotated by 45° to the horizontal and with the sun positioned vertically above the heliostat has been simulated in the STERG raytracer. For each ray increment the simulation was repeated $n = 10^5$ times. The mean value of each repeated simulation has been plotted with the vertical line representing the variance in the result. Fig. 2 shows that even for as few as 50 rays the long run mean solution is within 0.3 % of the analytical (deterministic) solution, although with a large degree of variance.

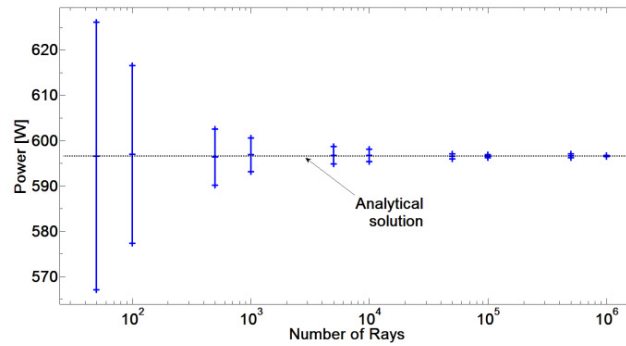


Fig 2. Results of a repeated simulation for various ray increments

In practice, a simulation can only be repeated a finite number of times, but if n is large enough a *point* estimate [12] of the mean μ_∞ , can be found from the standard arithmetic mean

$$\bar{y}_n = \frac{1}{n} \sum_{i=1}^n y_i \quad (1)$$

and the variance can be estimated by the sample standard deviation [11], using

$$s_n = \sqrt{\frac{1}{n-1} \left[\sum_{i=1}^n y_i^2 - \frac{(\sum_{i=1}^n y_i)^2}{n} \right]} \quad (2)$$

In Fig. 3 the frequency of the results using 50 rays for the same simulation (above) are plotted. For a large set of simulations the Central Limit Theorem holds [11] and the results of each trace are normally distributed about μ_∞ ,

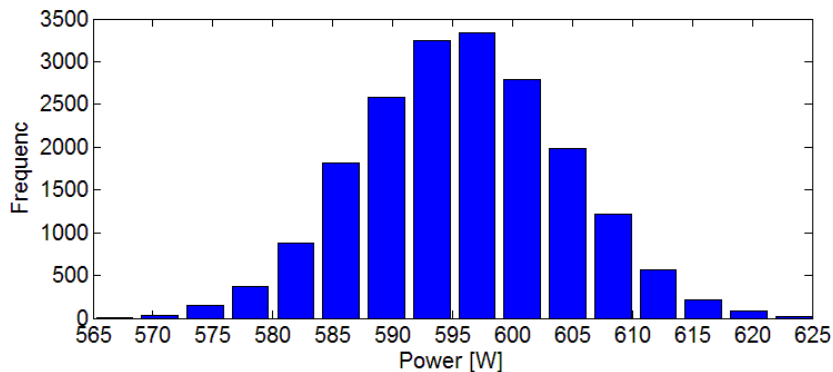


Fig 3. Frequency of results for multiple runs of a simulation

It is therefore possible to use a Gaussian distribution to set up an *interval* estimate of the mean μ_∞ . For a normal distribution approximately 99 % of the area under the curve lies within $z = 3$ standard deviations of μ_∞ . Therefore 99 % of the \bar{y} results will fall within the interval $\mu_\infty \pm z\sigma_N$. However, as it is unlikely that σ_N will be known nor is it practical to run a large number of simulations for s to be a reasonable estimate of σ_N . Therefore the more variable

students' t -distribution with $\nu = n - 1$ degrees of freedom is conservatively used [12]. The confidence interval estimate for μ_∞ is then

$$\bar{y} \pm t_\alpha \frac{s_n}{\sqrt{n}} \quad (3)$$

Where $(1-\alpha)$ is the coefficient which determines a $100(1-\alpha)$ % confidence interval for μ_∞ . This is a measure of goodness the interval estimate. The term on the right and side of the plus-or-minus is the error in the estimate.

$$E = t_\alpha \frac{s_n}{\sqrt{m}} \quad (4)$$

Three quantities influence the error and thus the width of the interval: the desired confidence level (which determines the t -value used), the standard deviation and the sample size. The confidence level, which is chosen by the user, needs to be carefully selected as a low value will mean that the within the interval there is likely going to be an error, in contrast computational cost of a high confidence may be prohibitive.

For a fixed confidence value the error can be reduced by increasing the number of rays. By rearranging Eq. (4) an estimate of the number of simulations required can be obtained.

$$m = \frac{t_\alpha^2 s_n^2}{E^2} \quad (5)$$

The value for m calculated in Eq. (5) is the number of times a simulation, using N rays needs to be repeated in order to resolve the solution to the simulation to within a tolerable error E , with a confidence of $100(1-\alpha)$ %. This is equivalent to running just one simulation with $m \cdot N$ rays and is the cornerstone of the algorithm.

3. The algorithm

As the STERG raytracer is primarily designed for use in a research institution the intent behind every simulation is likely to be different. For example, to test a field layout design, a user may wish to run a quick simulation to obtain an order of magnitude estimate of the total incident flux on a target, where as another user may wish to identify where the hotspots on a receiver are located and therefore requires a detailed flux map. The requirement of the simulation will affect the computational time of the trace where more precise spatially refined simulations require more rays and longer computational times. Accordingly the inputs into the STERG raytracer are not the number of rays to be traced but rather the level of accuracy or equivalently, the degree of error which a user is willing to accept for given spatial resolution.

The only unknown in Eq. (5) is the standard deviation. The first step in the algorithm is thus to obtain an estimate of s_n . This is done by initially tracing either $N_i = 50$ rays per object or $N_i = 50$ rays per division on the target, whichever is larger. The value of 50 has been empirically chosen as it was found to be a good balance of the least number of rays which can still provide a reasonable estimate of σ_n . The simulation is run twice ($n = 2$) and the standard deviation between the two simulations is calculated using Eq. (2).

The error is then calculated using Eq. (4). For a target subdivided into a number of cells the error for each cell is calculated. If this error equal or below the tolerable error the simulation stops. If however, in the more likely case, the error is greater than the tolerable error then Eq. (5) is used to predict the number of simulations required.

Finally the simulation is then run with a total of $m \cdot N$ rays. To further save on computational time the results of the initial two simulations are stored so that the final trace only requires $N(m - n)$ rays. The algorithm is summarized below:

1. **Set** Tolerable Error (E_T) and Confidence (α);
2. Run n simulations with $N=N_i \times \max(\text{Number of objects, Spatial Resolution})$;
3. Calculate the Error (E) using equation 2 and 4;
4. **If** $E_T < E$
 - a. Stop;
5. **Else**
 - a. Calculate the required number of simulations (m);
 - b. Run one simulation using $N(m - n)$ rays;

4. Results

4.1 Eurodish

To illustrate a practical implementation of the method and to serve as part validation of the STERG raytracer, the Eurodish has been simulated. The Eurodish has been used in a comparative validation study between SolTRACE and Tonatiuh [3] as well as SolFAST [5]. The simulation was run in the STERG raytracer using the same parameters for the Tonatiuh Eurodish geometry available online [13]. The results for total power on the receiver of the initial simulation are given in Table 1.

Table 1. Results of Eurodish for two simulations using 50 rays

Power Run 1	Power Run 2	Mean	Std. deviation	Error
57.126 kW	55.596 kW	56.361 kW	1.082 kW	3.118 kW

Using a tolerable error of 1 kW and a confidence interval of 98 % ($t = 2.353$), the total number of rays which need to be traced, from Eq. (5), is 350. Running the trace with a total of 350 rays the total power on the receiver is 55.43 kW which has a relative difference of 0.5 % for the same simulation run in Tonatiuh with 1mil rays. However, the radial distribution of rays is out by almost 8 % when compared to Tonatiuh, Fig. 4.

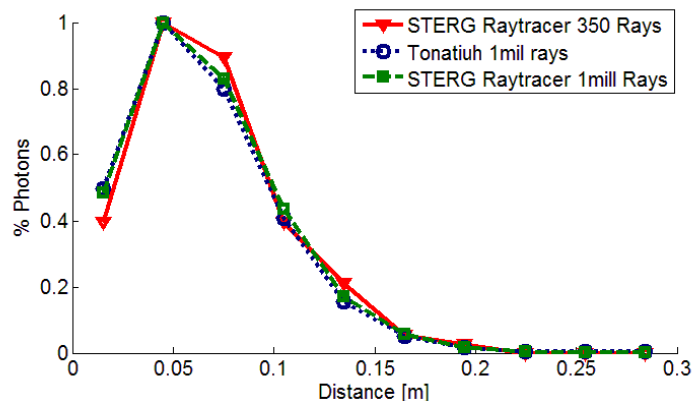


Fig. 4. Radial distribution of photons on Eurodish receiver

The results of the Eurodish test case show that very few rays can be used to predict the total incident power on the receiver albeit with poor spatial resolution.

4.2 Planta Solar 10

A second validation case was performed on an optimized field layout for the PS10 plant. The PS10 field has been used as a case study for a field optimization code developed by Lutchman [14]. Lutchman's code optimizes a field layout at hourly intervals for an entire year. The STERG raytracer was then used to spot check a large number of instantaneous points throughout the year. It is therefore crucial that no unnecessary rays are traced.

Presented here is a single case for the PS10 plant using the optimized field layout. The target has been discretised into a grid of 20×20 cells (each cell equal to $0.7 \text{ m} \times 0.7 \text{ m}$). A tolerable error of 100kW per cell or about 0.2% total intercepted power of PS10 at design point was chosen. The simulation was run twice with 20 000 rays. The power, which is calculated from the number of ray hits per cell, and error, which is calculated from Eq. (2) are illustrated in Fig 5. As can be seen in Fig.5(b), in numerous cells the error exceeds 100 kW with a maximum error of 953.98 kW.

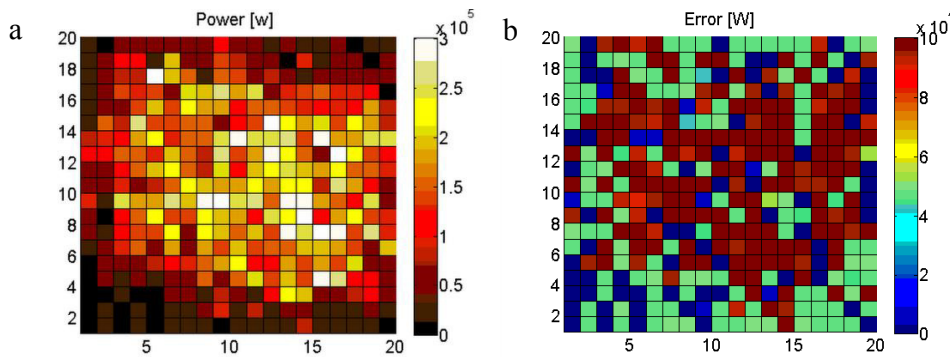


Fig. 5 Optimized PS10 using 20 000 (a) The total intercepted power per cell (b) the total error per cell

As the maximum error exceeds the tolerable error, the STERG raytracer then uses Eq. (5) to predict and trace the required number of rays. The total number of rays required to reduce the total error below the tolerable error with a 98 % confidence is $m = 320\,850$. To substantiate this, the results of a simulation run with $m/2$ rays is presented in Fig. 6. From Fig 6(b) the error in all cells is below the required 100 kW with a maximum error of only 50.737 kW.

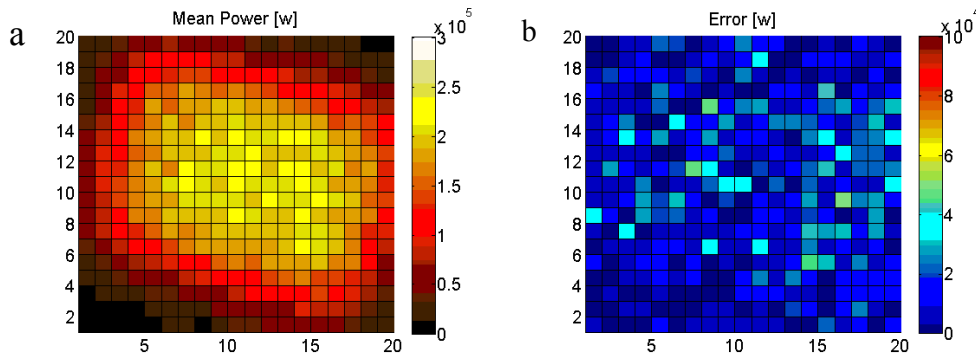


Fig. 6. Optimized PS10 using 320 850 (a) The total mean intercepted power on the target (b) Total error per cell on the target

5. Discussion

Monte Carlo ray tracing is a powerful tool to aid in the simulation and design of CSP systems. The method presented here has been implemented in the STERG raytracer to assist a non-optics researcher in ray tracing. It does this by using a novel approach of shifting the inputs from number of sampled rays to the more familiar concepts of

tolerable error and confidence.

The default settings of the STERG raytracer are to run only two initial simulations with as few as 50 rays per object. With so few rays the results of the initial simulations will have a great deal of variability. Due to the stochastic generation of rays, it is possible that the results of the two simulations are very similar. However the coverage factor (the t -value) is a multiplying factor which ensures a larger interval estimate which encompasses most possible results (depending on the confidence). Conversely, if the two simulations produce results which are on the extremes of the range, the method will over predict the number of rays required. It was found however, that these two situations are rare and for most validation cases, which include the simulation of the Eurodish and optimized PS-10 plant presented here as well as the experimental validation using an actual heliostat, the method suitably predicts the required number of rays.

Although the method presented here aims to better assist a user of the STERG raytracer, as with all numerical simulation tools, a certain degree of insight of the simulation is required. A user should take care when defining the tolerable error and confidence intervals as, for example, defining a tolerable error of a few watts when the incident power on the target is in the order of megawatts requires a prohibitively long computational time.

6. Conclusions

A method developed and implemented into a raytracer developed at STERG which estimates the number of rays required in a simulation has been presented. The inputs into the method are tolerable error and confidence of the result. By running two initial simulations point estimates of the mean and variance of the result is obtained as well the error in the result. Using these values the ray tracer uses Eq. (5) to estimate the required number of rays to reduce the error to a value below the tolerable error defined by the user. For all validation cases the method adequately predicted the required number of rays.

Acknowledgements

The authors would like to thank Sasol Technology, the Department of Science and Technology of South Africa through the Solar thermal spoke fund and the Stellenbosch University Hope project for funding the resources to perform this work and present it at SolarPACES.

References

- [1] Bode S, and Gauche P, "Review of optical software for use in concentrating solar power systems," in South African Solar Energy Conference, Stellenbosch, 2012
- [2] Ferriere A, Bezier JJ, Garcia P, "Codes for solar flux calculations dedicated to central receiver system applications: A comparative review," Solar Energy, vol. 82, pp. 189-197, 2008.
- [3] Blanco MJ, Mutuberria A, Garcia P, Gastesi R, and Martiv V, Preliminary validation of Tonaituih, SolarPACES Symposium, Berlin, Germany 2009
- [4] Wendelin T. Soltrace: A new optical modelling tool for concentrating solar optics ASME International Solar Energy Conference, Kohala Coast, Hawaii, 2003
- [5] Roccia JP, Piaud B, Coustet C, Caliot C, Guilliot E, Flamant F, Deltorre J, SOLFAST, a Ray-Tracing Monte Carlo software for solar concentrating facilities, , Journal of Physics conference series. 2012, 369, 012029
- [6] Izigon M, Armstrong P, Nilsson C and Vu N, TieSOL – A GPU based suite of software for central receiver power plants, SolarPACES Granada, Spain, 2011
- [7] Delatorre, J., et al. "Monte Carlo advances and concentrated solar applications." Solar Energy (2013).
- [8] Arvo J and Kirk D, An introduction to ray tracing: A survey of ray tracing acceleration technique, 1989, 201-262
- [9] Pharr M, Humphreys G, Physically Based Rendering: From theory to implantation, 2nd ed. Morgan Kaufmann 2010
- [10] Gardiner CW, Handbook of Stochastic Methods: For Physics, Chemistry and Natural Science 3rd ed., Springer 1985
- [11] Lyman Ott R, An Introduction to Statistical Methods and Data Analysis, 4th ed. Duxbury Press, 1984
- [12] Kroese DP, Taimre T, Botev ZI, Handbook of Monte Carlo Methods, John Wiley & Sons 2011
- [13] Mutuberria A, ParabolicDish.tnh, Available at: <https://code.google.com/p/tonaituih/downloads/detail?name=ParabolicDish.tnh>, Last accessed: 1 July 2013
- [14] Lutchman SL, Personal Communication, Stellenbosch 2013